Inferential Statistics in R

Alessio Palmisano

1. Linear Regression

In this class, you will examine the distribution of Roman pottery across central and southern England using linear regression. A similar analysis was discussed in Hodder and Orton's (1976: 115–119) foundational work on spatial analysis in archaeology.

For today's tutorial, you will work with four vector data:

1. **sites**: contains the locations of 30 Romano-British sites, along with an attribute table showing the percentage of Oxford pottery (**oxpots**), the percentage of New Forest pottery (**nfpots**), and whether the site has a transport link to Oxford (**transport**).
2. **oxford**: the location of the Oxford kiln.
3. **newforest**: the location of the New Forest kiln.
4. **uk**: A vector polygon map representing the United Kingdom.

The of this study is to analyse the variation in the proportion of Oxford pottery found at major Romano-British sites in southeastern England. It also seeks to examine whether this variation is linked to the distance between these sites and the Oxford and New Forest kilns.

Now, set your working directory and import all the vector data above.

Before doing so, install the package "sf" by typing the following command:

```
install.packages("sf")
```

The sf package (short for Simple Features) is the modern, efficient way to handle spatial data in R.

```
library(sf) #Load the package

## Warning: package 'sf' was built under R version 4.3.3

## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

Load the vector data by setting your working directory as the "data" folder of this class:

```
uk <- st_read("linear_regression/uk.shp")

## Reading layer `uk' from data source
##   `/Users/alessio/Desktop/Quantifying Culture/corso/class 3/data/linear_re
gression/uk.shp'
```

```
##   using driver `ESRI Shapefile'
## Simple feature collection with 147 features and 1 field
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 92664.32 ymin: 11700.93 xmax: 655327.7 ymax: 1056325
## Projected CRS: OSGB36 / British National Grid
```

```
sites<-st_read("linear_regression/sites.shp")
```

```
## Reading layer `sites' from data source
##   `/Users/alessio/Desktop/Quantifying Culture/corso/class 3/data/linear_re
gression/sites.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 30 features and 6 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 347030.1 ymin: 92121.2 xmax: 649192.8 ymax: 432060.1
## Projected CRS: Transverse_Mercator
```

```
oxford<-st_read("linear_regression/oxford.shp")
```

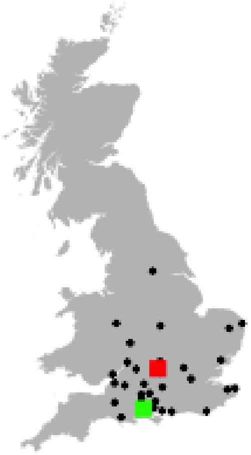
```
## Reading layer `oxford' from data source
##   `/Users/alessio/Desktop/Quantifying Culture/corso/class 3/data/linear_re
gression/oxford.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 1 feature and 2 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 452643.6 ymin: 206020.4 xmax: 452643.6 ymax: 206020.4
## Projected CRS: Transverse_Mercator
```

```
newforest<-st_read("linear_regression/newforest.shp")
```

```
## Reading layer `newforest' from data source
##   `/Users/alessio/Desktop/Quantifying Culture/corso/class 3/data/linear_re
gression/newforest.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 1 feature and 2 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 419205.3 ymin: 111540.2 xmax: 419205.3 ymax: 111540.2
## Projected CRS: Transverse_Mercator
```

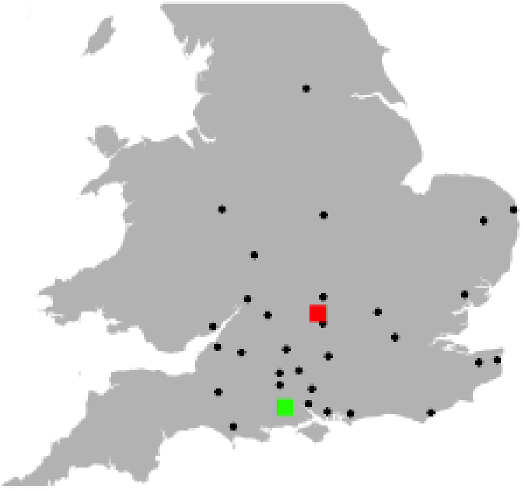
Now we loaded all the spatial data, Let us plot the spatial data we have:

```
plot(st_geometry(uk), col="grey75", border=NA)
points(st_coordinates(sites), pch=19, cex=0.3)
points(st_coordinates(newforest), col="green", pch=15)
points(st_coordinates(oxford), col="red", pch=15)
```



Given that the resulting map is too small, we can adjust the `xlim` and `ylim` parameters in plot(). Therefore, you can zoom in by adjusting the limits slightly:

```
plot(st_geometry(uk), col="grey75", border=NA,  xlim=c(300000, 650000), ylim=
c(50000, 500000))
points(st_coordinates(sites), pch=19, cex=0.3)
points(st_coordinates(newforest), col="green", pch=15)
points(st_coordinates(oxford), col="red", pch=15)
```



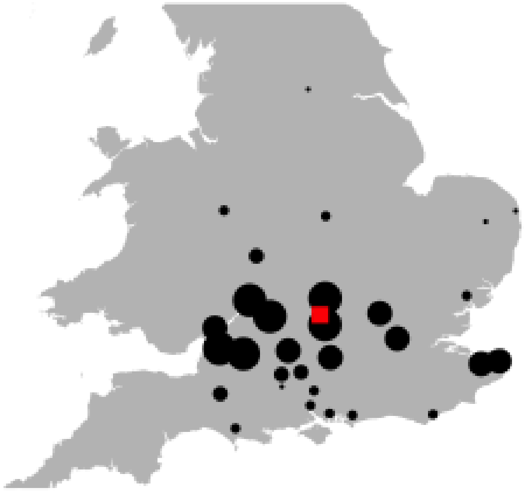
The archaeological sites containing Oxfordshire and New Forest pottery are represented by black circles, while the New Forest kilns are marked with a green square and the Oxford kilns with a red square.

Let us explore some descriptive statistics about the Oxford and New Forest pottery percentage found in the sites:

```
summary(sites)

##       cat              Id      site_name          transport
##  Min.   : 1.00   Min.   :0   Length:30          Length:30
##  1st Qu.: 8.25   1st Qu.:0   Class :character   Class :character
##  Median :15.50   Median :0   Mode  :character   Mode  :character
##  Mean   :15.50   Mean   :0
##  3rd Qu.:22.75   3rd Qu.:0
##  Max.   :30.00   Max.   :0
##      oxpots           nfpots                geometry
##  Min.   : 1.500   Min.   : 0.000   POINT        :30
##  1st Qu.: 6.812   1st Qu.: 0.000   epsg:NA      : 0
##  Median :11.750   Median : 0.000   +proj=tmer...: 0
##  Mean   :12.712   Mean   : 4.202
##  3rd Qu.:18.812   3rd Qu.: 8.062
##  Max.   :22.500   Max.   :17.500
```

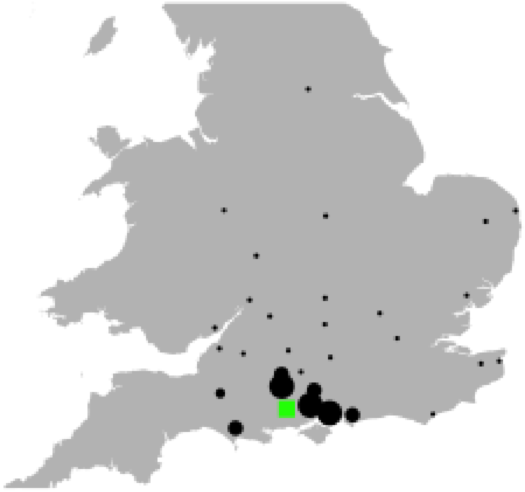
The percentages in the “oxpots” and “nfpots” columns range roughly from 0 to 25, representing the proportion of the total Roman pottery assemblage at each site. To visualize this thematically, we can create a simple plot using 5 graduated symbol classes by categorizing different proportions and representing them as circles of varying sizes.

```
plot(st_geometry(uk), col="grey75", border=NA,  xlim=c(300000, 650000), ylim=
c(50000, 500000))
points(st_coordinates(sites[sites$oxpots >= 0 & sites$oxpots <= 5,]), pch=16,
cex=0.3)
points(st_coordinates(sites[sites$oxpots >= 5 & sites$oxpots <= 10,]), pch=16
, cex=0.6)
points(st_coordinates(sites[sites$oxpots >= 10 & sites$oxpots <= 15,]), pch=1
6, cex=0.9)
points(st_coordinates(sites[sites$oxpots >= 15 & sites$oxpots <= 20,]), pch=1
6, cex=1.5)
points(st_coordinates(sites[sites$oxpots >= 20 & sites$oxpots <= 25,]), pch=1
6, cex=2)
points(st_coordinates(oxford), col="red", pch=15)
```



Now, you can do the same plot for the New Forest data.

```
plot(st_geometry(uk), col="grey75", border=NA,  xlim=c(300000, 650000), ylim=
c(50000, 500000))
points(st_coordinates(sites[sites$nfpots >= 0 & sites$nfpots <= 5,]), pch=16,
cex=0.3)
points(st_coordinates(sites[sites$nfpots >= 5 & sites$nfpots <= 10,]), pch=16
, cex=0.6)
points(st_coordinates(sites[sites$nfpots >= 10 & sites$nfpots <= 15,]), pch=1
6, cex=0.9)
points(st_coordinates(sites[sites$nfpots >= 15 & sites$nfpots <= 20,]), pch=1
6, cex=1.5)
points(st_coordinates(sites[sites$nfpots >= 20 & sites$nfpots <= 25,]), pch=1
6, cex=2)
points(st_coordinates(newforest), col="green", pch=15)
```



Next, we need to calculate the distance from the two pottery production centers (Oxford and New Forest) to each of the observed archaeological sites.

```
coord<- rbind(st_coordinates(sites),st_coordinates(newforest),st_coordinates(
oxford))
dist_matr<- as.matrix(dist(coord))
sites$distox <- dist_matr[32,1:30]
sites$distnf <- dist_matr[31,1:30]
```

The current distance is measured in meters (as per the coordinate system's units), so let's convert these values to kilometers.

```
sites$distox <- sites$distox / 1000
sites$distnf <- sites$distnf / 1000
```

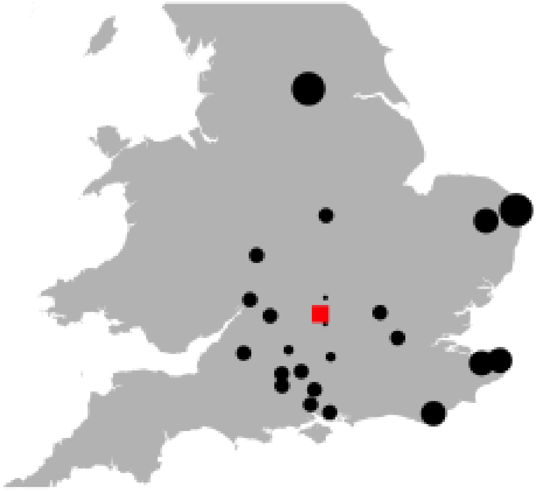
You might want to verify the accuracy of the distances visually, for example, by plotting them thematically using graduated symbols, similar to the method you applied earlier.

First, let us check some descriptive statitics about the distance of our sites from oxford:

```
summary(sites$distox)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   11.05   71.48   95.24  105.42  142.18  226.35 
```

Now, plot the distance from Oxford by using graduate colors:

```
plot(st_geometry(uk), col="grey75", border=NA,  xlim=c(300000, 650000), ylim=
c(50000, 500000))
points(st_coordinates(sites[sites$distox >= 0 & sites$distox <= 25,]), pch=16
, cex=0.3)
points(st_coordinates(sites[sites$distox >= 25 & sites$distox <= 50,]), pch=1
6, cex=0.6)
points(st_coordinates(sites[sites$distox >= 50 & sites$distox <= 100,]), pch=
16, cex=0.9)
points(st_coordinates(sites[sites$distox >= 150 & sites$distox <= 200,]), pch
=16, cex=1.5)
points(st_coordinates(sites[sites$distox >200,]), pch=16, cex=2)
points(st_coordinates(oxford), col="red", pch=15)
```

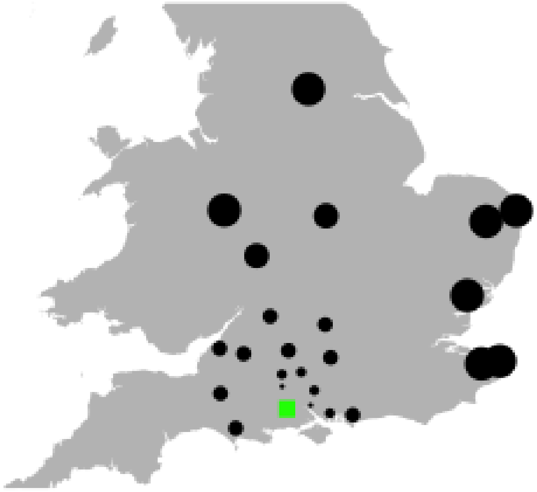


Let us do the same for the distances from New Forest:

```
summary(sites$distnf)
```

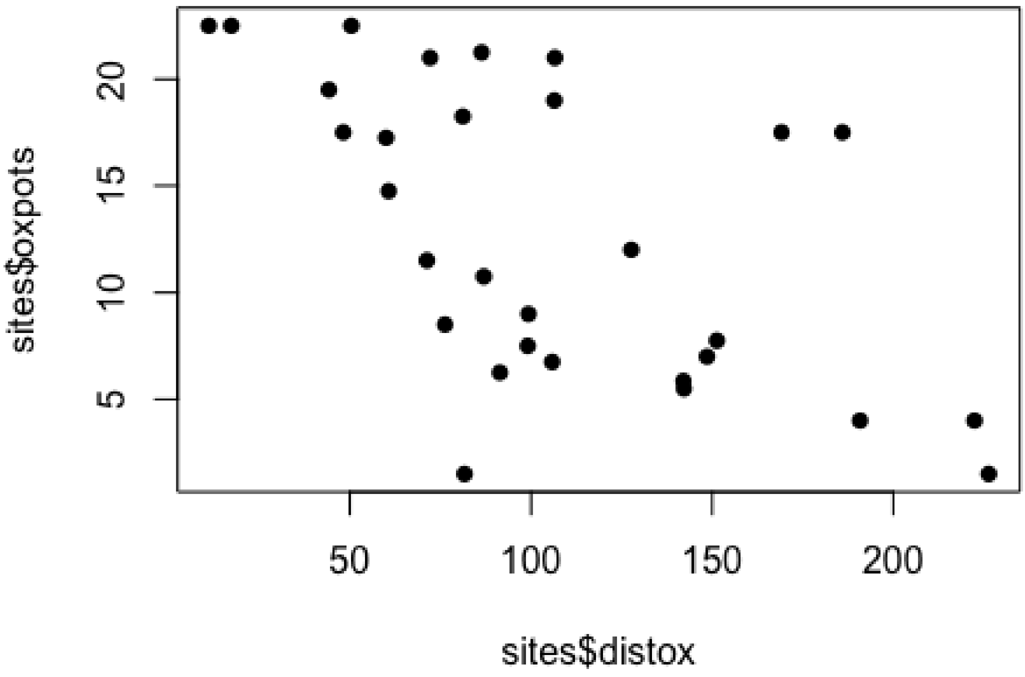
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   22.99   60.39  101.47  123.67  187.10  321.24
```

```
plot(st_geometry(uk), col="grey75", border=NA,  xlim=c(300000, 650000), ylim=
c(50000, 500000))
points(st_coordinates(sites[sites$distnf >= 0 & sites$distnf <= 25,]), pch=16
, cex=0.3)
points(st_coordinates(sites[sites$distnf >= 25 & sites$distnf <= 50,]), pch=1
6, cex=0.6)
points(st_coordinates(sites[sites$distnf >= 50 & sites$distnf <= 100,]), pch=
16, cex=0.9)
points(st_coordinates(sites[sites$distnf >= 150 & sites$distnf <= 200,]), pch
=16, cex=1.5)
points(st_coordinates(sites[sites$distnf >200,]), pch=16, cex=2)
points(st_coordinates(newforest), col="green", pch=15)
```



Now we can finally compute a linear regression. First, let us assess the distance from oxford and the percentage of pottery by using just a simple scatterplot:

```r
plot(sites$distox,sites$oxpots, pch=16)
```



From the graph above, we can observe a clear relationship between the percentage of Oxfordshire pottery found at each Roman site and its distance from the production center in Oxford. To formalize this relationship, we can apply a linear regression model, using the distance from the Oxfordshire production center as a predictor for the proportion of Oxfordshire pottery present at each site.

Therefore, we will fit a simple linear regression model, where the dependent variable is the percentage of pottery (*oxpots*), and the independent variable (or covariate) is the distance from Oxford (*dist_ox*).

Mathematically, this will is represented by the following equation:

$$y \sim \mathcal{N}(\alpha + \beta x, \epsilon)$$

where:

- α is the **intercept** (the value of *y* when *x* = 0).
- β is the **slope** (how much *y* changes for a one-unit increase in *x*).
- x is the **input variable** (independent variable or predictor).
- ϵ represents the **standard deviation** of the normal distribution, capturing the random noise or variability in *y* that is not explained by the linear relationship.

The equation states that for a given *x*, the output *y* is normally distributed around $\alpha + \beta x$ with a spread (variance) controlled by ϵ.

We can implement the equation above in R as follows:

```
lregr <- lm(sites$oxpots ~ sites$distox)
```

We can also calculate the Pearson *r* correlation between the quantity of pottery and the distance from the production centre:

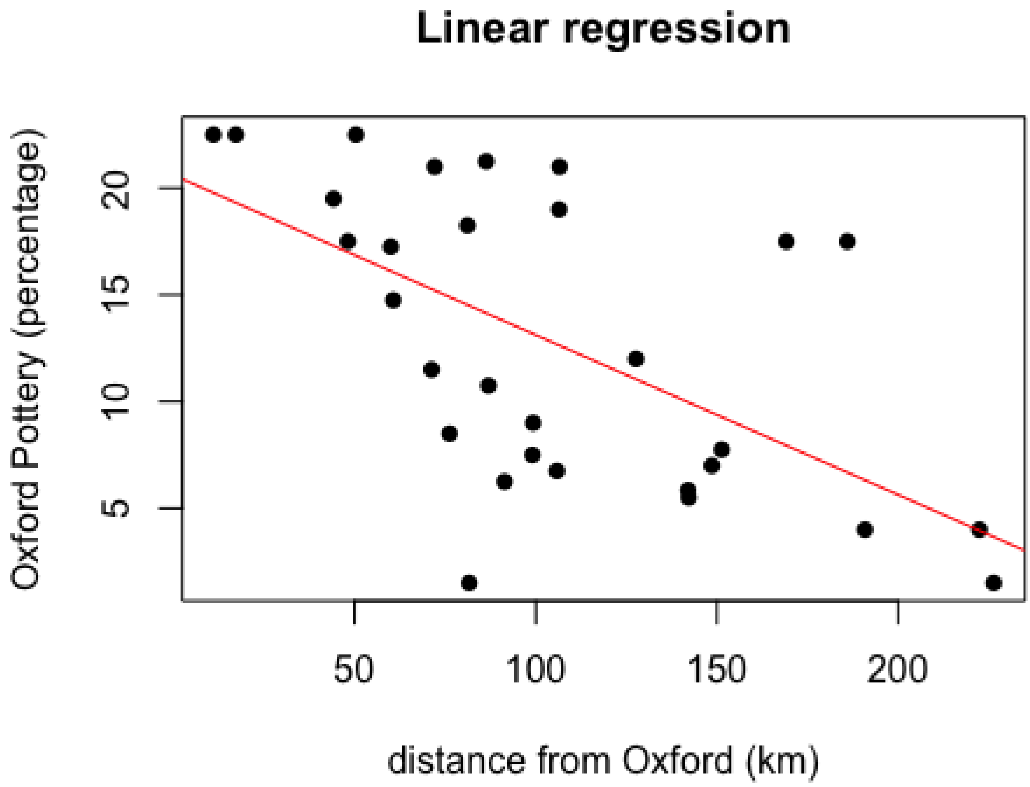
```
cor(sites$oxpots,sites$distox)

## [1] -0.5965397
```

The result indicates that the two variables are negatively correlated.

The `lm()` function enables us to create a linear model, which will be stored in the object lregr.

We will now plot the linear regression in red:

```
plot(sites$distox,sites$oxpots, main="Linear regression", xlab = "distance fr
om Oxford (km)",  ylab="Oxford Pottery (percentage)", pch=16)
abline(lregr, col="red")
```



This will provide a visual representation of the fitted model. Do you think it accurately represents the data?

To further assess the regression analysis, we can use the summary() function. This function will generate statistical details related to the regression, allowing us to evaluate the model's performance and significance.

```
summary(lregr)

##
## Call:
## lm(formula = sites$oxpots ~ sites$distox)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9931  -4.0310  -0.6302   3.5874  10.8196
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.60720    2.26062   9.116 7.13e-10 ***
## sites$distox -0.07490    0.01904  -3.933 0.000503 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 5.693 on 28 degrees of freedom
## Multiple R-squared:  0.3559, Adjusted R-squared:  0.3329
## F-statistic: 15.47 on 1 and 28 DF,  p-value: 0.0005027
```

Here the explanation of the summary of your linear regression.

- **Residuals**: provide a basic statistical summary of the distribution of the differences between the observed and predicted values. This summary includes key metrics such as the minimum, maximum, first quartile (Q1), third quartile (Q3), and the median. These statistics give an indication of how well the model fits the data by highlighting the spread and central tendency of the residuals. A well-fitting model typically has residuals that are symmetrically distributed around zero, with no significant outliers.

- **Coefficients** section provides detailed information about the intercept and the coefficients associated with the covariates in the regression model.

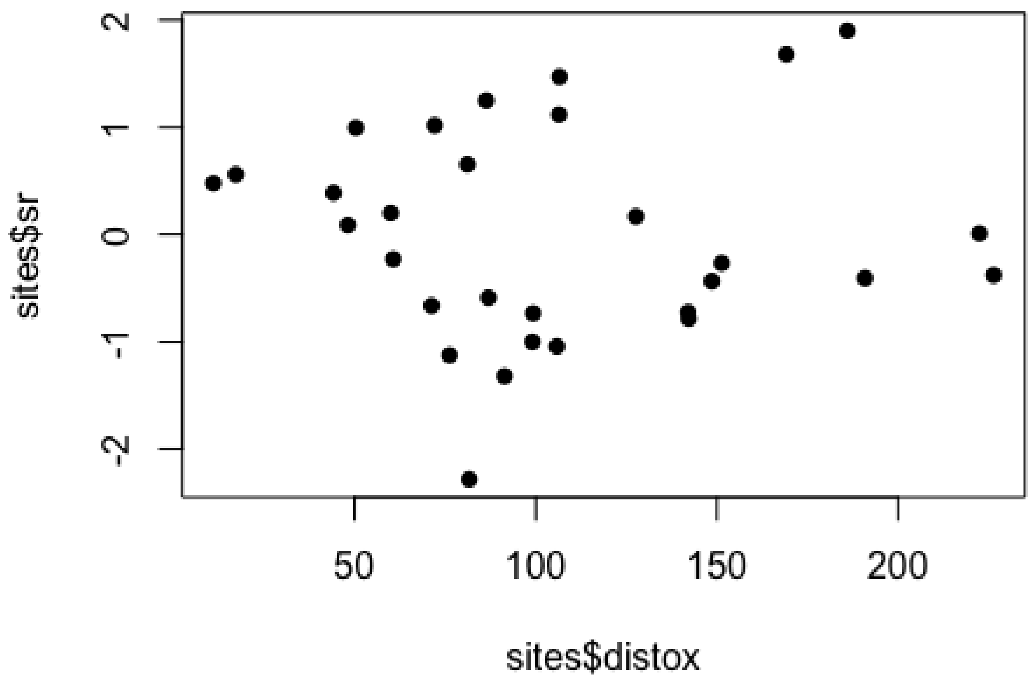
  - *Estimate of the intercept and coefficient(s)*: These are the values that define the relationship between the dependent and independent variables. In our case, the intercept (20.60720) represents the predicted value of the dependent variable when sites$distox = 0.
  - *Standard error*: This measures the variability of the coefficient estimates.The standard error is 2.26062, indicating the uncertainty in estimating the intercept.
  - *t-value*: The ratio of the coefficient to its standard error, which helps assess the significance of the coefficient. A larger t-value typically indicates that the coefficient is more likely to be significantly different from zero. The t-value of 9.116 is very high, meaning the intercept is statistically significant.
  - *p-value*: This is the result of a t-test conducted for each coefficient. A small p-value (typically less than 0.05) suggests that the coefficient is statistically significant.
  - *Slope*: This means that for every one-unit increase in sites$distox, the response variable decreases by 0.0749 on average.
- **Multiple R-squared**: indicates the strength of the relationship between the dependent and independent variables in the regression model. A higher R^2 value suggests a stronger relationship.This indicates that 35.59% of the variability in the dependent variable is explained by sites$distox.

Examining the residuals more closely can offer valuable insights into the reliability of the regression model. One effective method is to plot the standardized residuals against the covariate or the predicted values. To start, we will calculate the standardized residuals by dividing the residuals (which we can extract from the model using the `residuals()` function) by the residual standard error (which is available through the `summary()` function). These standardized residuals will then be added to a new column in the attribute table of the sites:

```
sites$sr <- residuals(lregr) / 5.693
```

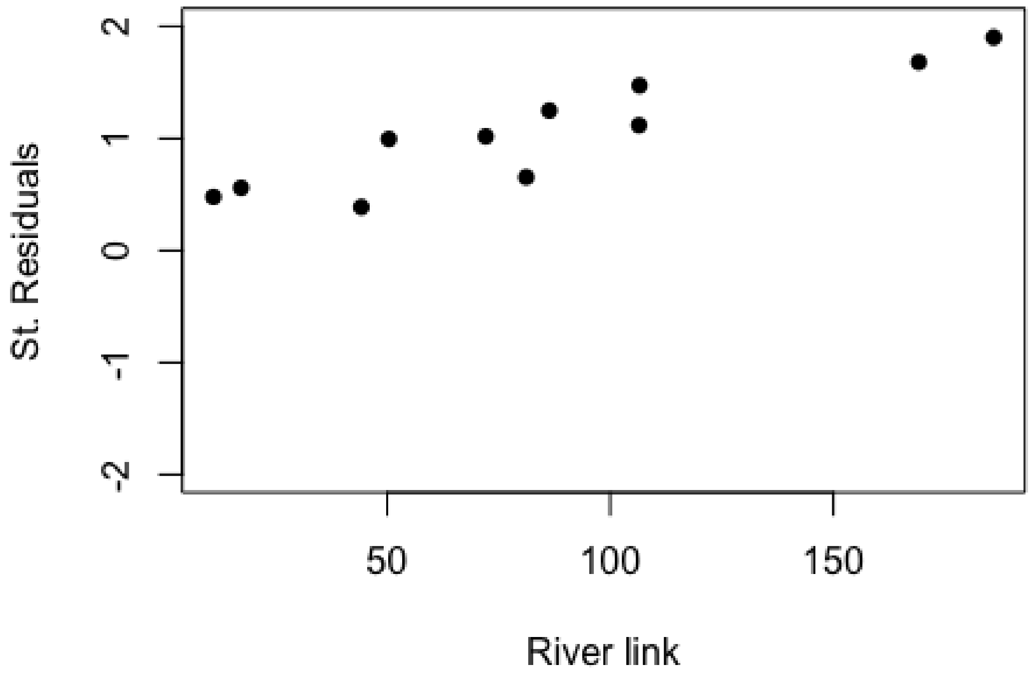
Next, we will create a plot of the standardized residuals against the covariate:

```
plot(sites$distox,sites$sr, pch=16)
```



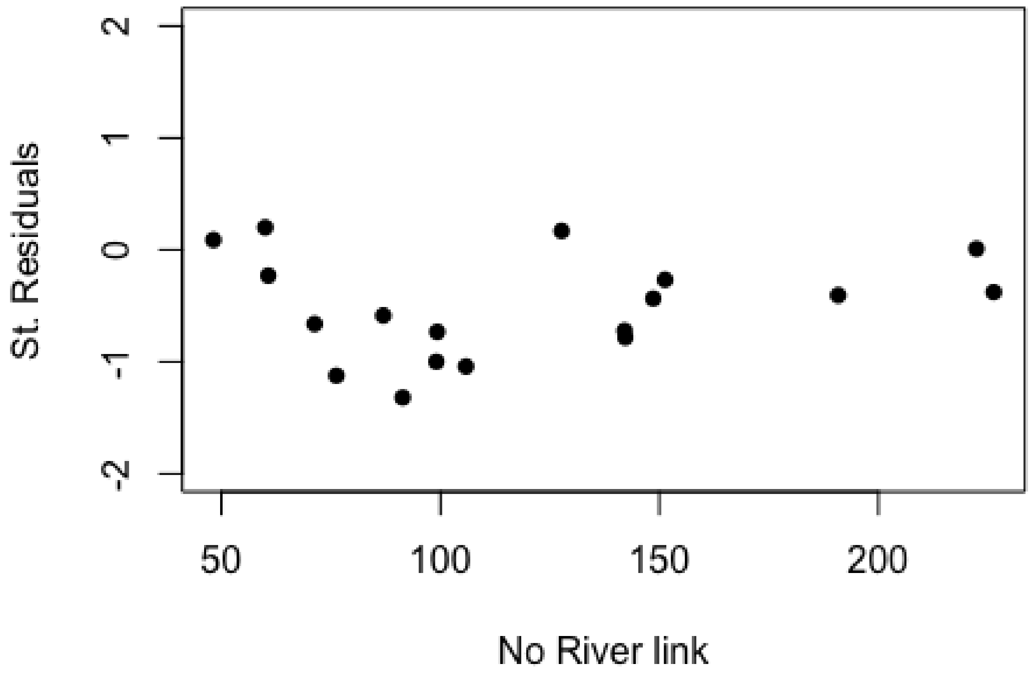
The scatterplot should ideally appear random. If the regression model accurately represents the relationship between the dependent and independent variables, the residuals should reflect random errors. However, if a systematic pattern emerges, it suggests that there may be additional factors not accounted for in the model, indicating that further investigation is needed. The dataset also includes one additional variable, *transport*, which indicates the presence (TRUE) or absence (FALSE) of a river-based transport link between each site and Oxford. Let's explore how the standardized residuals differ based on the value of this variable:

```
plot(sites$distox[sites$transport==TRUE],sites$sr[sites$transport == TRUE], y
lab="St. Residuals", xlab="River link", ylim = c(-2,2), pch=16)
```



Now, let us do the oppsite by plotting the residuals of those site not having a river-based transport link:

```
plot(sites$distox[sites$transport==FALSE],sites$sr[sites$transport == FALSE],
ylab="St. Residuals", xlab="No River link", ylim = c(-2,2), pch=16)
```



Do you notice difference between those two graphs?

Considering the noticeable correlation between the residuals and whether a river-based transport link to Oxford exists, it seems logical to develop two distinct regression models.

First, we will divide the sites into two groups: those with a transport link and those without.

```
t.regdata <- sites[sites$transport==TRUE,]
nt.regdata <- sites[sites$transport==FALSE,]
```

We now have two separate data frames: one containing the sites with a water-based transport link to Oxford (*t.regdata*), and the other containing the sites without such a link (*nt.regdata*)

We can now easily compute the regression for the two subsets we just created:

```
lregr.t <- lm(oxpots~distox, data=t.regdata)
lregr.nt <- lm(oxpots~distox, data=nt.regdata)
```

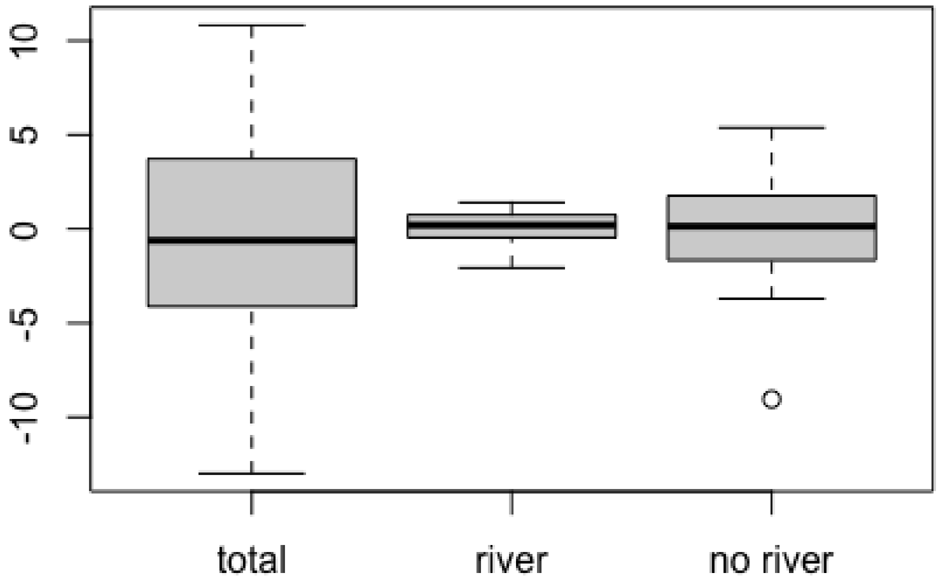
Examine the statistics of the two linear regressions:

```
summary(lregr.t)

##
## Call:
## lm(formula = oxpots ~ distox, data = t.regdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0747 -0.4706  0.1895  0.7455  1.3949
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.623546   0.691015  32.740 1.14e-10 ***
## distox      -0.028324   0.006909  -4.099  0.00268 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.222 on 9 degrees of freedom
## Multiple R-squared:  0.6512, Adjusted R-squared:  0.6125
## F-statistic: 16.81 on 1 and 9 DF,  p-value: 0.002679

summary(lregr.nt)

##
## Call:
## lm(formula = oxpots ~ distox, data = nt.regdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0511 -1.6664  0.1244  1.7557  5.3756
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.54112    1.95917   7.932  4.1e-07 ***
## distox      -0.06112    0.01527  -4.003 0.000921 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.433 on 17 degrees of freedom
## Multiple R-squared:  0.4852, Adjusted R-squared:  0.4549
## F-statistic: 16.02 on 1 and 17 DF,  p-value: 0.0009215
```

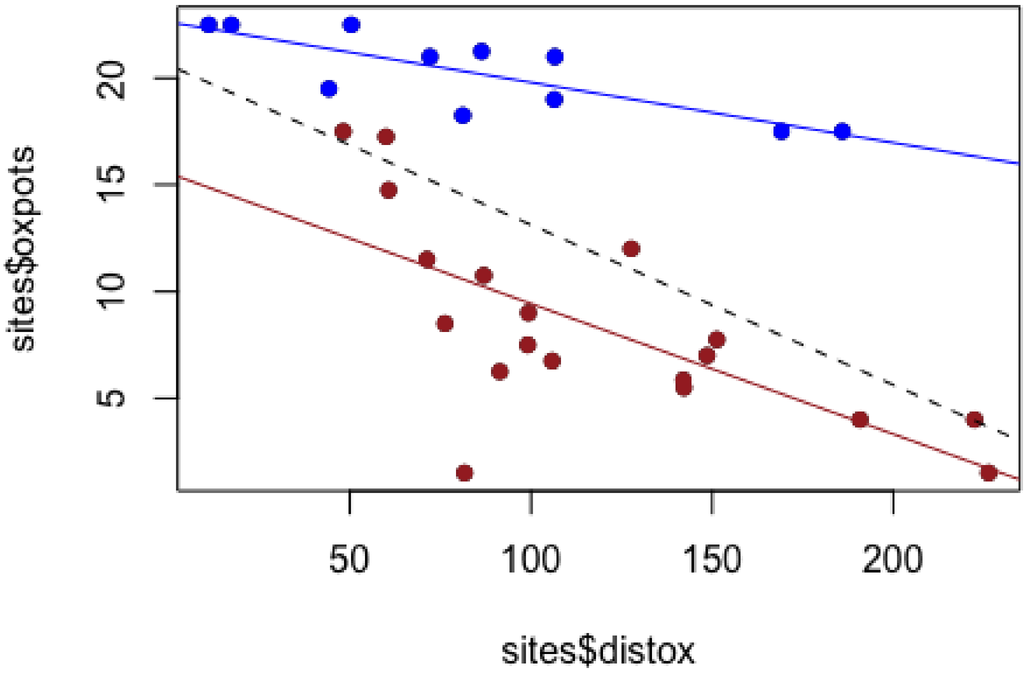
The improvement of the two new models compared to the original model is evident in the decreased variability of the residuals. We can visually evaluate this by comparing boxplots of the residuals

```
total <- residuals(lregr)
transport <- residuals(lregr.t)
no_transport <- residuals(lregr.nt)
boxplot(total,transport,no_transport, names=c("total","river","no river"))
```



Finally, we visualize all three models in a single plot, using distinct colors to represent each model:

```
plot(sites$distox,sites$oxpots, pch=16)
points(t.regdata$distox,t.regdata$oxpots,col="blue",pch=16)
points(nt.regdata$distox,nt.regdata$oxpots,col="brown",pch=16)
abline(lregr,lty="dashed",col="black")
abline(lregr.t,col="blue")
abline(lregr.nt,col="brown")
```



The dashed line represents the global regression model, while the blue and brown lines correspond to the models for sites with and without a river-based transport link, respectively

2. Chi-square test

The Chi-Squared (χ^2) Test is a statistical test used to determine if there is a significant association between two categorical variables. It helps assess whether observed frequencies differ from expected frequencies due to chance or if there is a real relationship between the variables.

In this tutorial we will explore the Chi-square test for independence, which is used to test if two categorical variables are related.

Imagine a region characterised by three distinct soil types: alluvial, loamy, and rocky. Within this area, 280 Bronze Age settlements have been identified. An initial examination of the map suggests that settlements may have been more frequently located on alluvial and loamy soils. The key question is whether this pattern occurred by chance or if there was a preference for certain soil types.

If all three soil types were equally suitable for settlement, one would expect a relatively uniform distribution of settlements across the landscape. In this case, any variations in settlement density would likely be attributed to minor differences in topography or individual choices made by the settlers, rather than soil composition. To assess whether the observed distribution is statistically significant or merely random, the chi-squared test can be applied.

The first step is to define the hypotheses:

1. **Null Hypothesis (H_0)**: Settlement type (Rural or Urban) is independent of soil type (i.e., soil type does not influence settlement distribution).
2. **Alternative Hypothesis (H_1)**: Settlement type is dependent on soil type (i.e., certain settlements are more likely to be found on specific soils).

Set the working directory and let us import the dataset:

```
data<-read.csv(file="chi-square test/settlements_soil.csv", header = TRUE, se
p=",")
```

Now, print the observed data:

```
data

##          X Rural Urban
## 1 Alluvial    70    40
## 2    Loamy    90    30
## 3    Rocky    50     0
```

To calculate the theoretically derived expected frequencies, we first need to assume that the distribution of settlements across the three soil types (alluvial, loamy and rocky) is random. In other words, if there is no preference for any particular soil type.

The expected frequencies can be computed using the formula: $E = \frac{(\text{Row Total} \times \text{Column Total})}{\text{Grand Total}}$

Let us perform it in R:

```
# Total number of settlements
total_settlements <- sum(data[,2:3])

# Row totals (for each soil type)
row_totals <- rowSums(data[,2:3])

# Column totals (Rural and Urban settlements)
```

```r
col_totals <- colSums(data[,2:3])

# Compute expected frequencies
expected <- outer(row_totals, col_totals) / total_settlements

# Print the expected frequencies
expected

##      Rural Urban
## [1,]  82.5  27.5
## [2,]  90.0  30.0
## [3,]  37.5  12.5
```

Now, let us compute the chi-square test:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

In this formula, k represents the number of categories, O_i denotes the observed frequency in category i, E_i stands for the expected frequency in category i, and χ^2 symbolizes the chi-squared statistic, using the Greek letter ‘chi’.

or Alluvial & Rural:

$$\frac{(70 - 82.5)^2}{82.5} = \frac{(-12.5)^2}{82.5} = \frac{156.25}{82.5} = 1.894$$

An then you do so for each categories, such as Loamy & Rural and so on..

Now, let’s perform the Chi-Square Test using the **chisq.test()** function in R, which will calculate the Chi-Square statistic and the p-value for us.

```r
# Perform Chi-Square Test
chi_square_test <- chisq.test(data[,2:3])

# Print the Chi-Square Test results
chi_square_test

##
##  Pearson's Chi-squared test
##
## data:  data[, 2:3]
## X-squared = 24.242, df = 2, p-value = 5.443e-06
```

The chi-square test provide us with the *Chi-squared* statistics (24.242), the *degree of freedom* (df=2), and the *p-value*.

Since χ^2 = 24.24 is much greater than the critical value (5.99 for df = 2 at α = 0.05), and the p-value is very small (0.00000544), we reject the null hypothesis.

This confirms that settlement type is **strongly dependent on soil type**. The fact that no urban settlements were found in rocky soils is statistically significant, showing that people likely avoided urbanizing in difficult terrains.

3. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) simplifies a dataset by identifying variables that strongly correlate, suggesting they reflect the same underlying factor. Instead of retaining all variables, PCA replaces them with a smaller set of components while preserving key relationships within the data. Unlike trial-and-error methods, PCA mathematically extracts principal components by analyzing the correlation matrix. The aim is to create a minimal set of components that maintain strong associations with the original variables.

Today we will perform PCA by using the dataset sourced from Tubb *et al.* (1980), which includes 48 rows (cases) and 12 columns (variables). The columns from 4 to 12 contain oxide concentration percentages in Romano-British pottery samples, forming a 48×9 data matrix for analysis. The second and third columns records respectively the kiln site and region where each specimen was found and is used for labeling plots.

Therefore, set the directory and load the dataset stored within the folder data/PCA

```
# Step 1: Load the dataset
pottery<-read.csv(file="PCA/pottery.csv", header = TRUE, sep = ",")
```

The questions that were raised in the original paper by Tubb and colleagues were the following:

- Is there evidence of chemical grouping in the data?
- If grouping occurs can it be associated with the region of provenance?

So, if grouping is observed the next step is to investigate if it correlates with the region of provenance. A strong association would suggest that that pottery samples from the **same region share similar chemical compositions**, potentially indicating a link between **production techniques** or **raw material sources**.

So, let u now compute the covariance matrix, which shows how variables in the dataset are correlated:

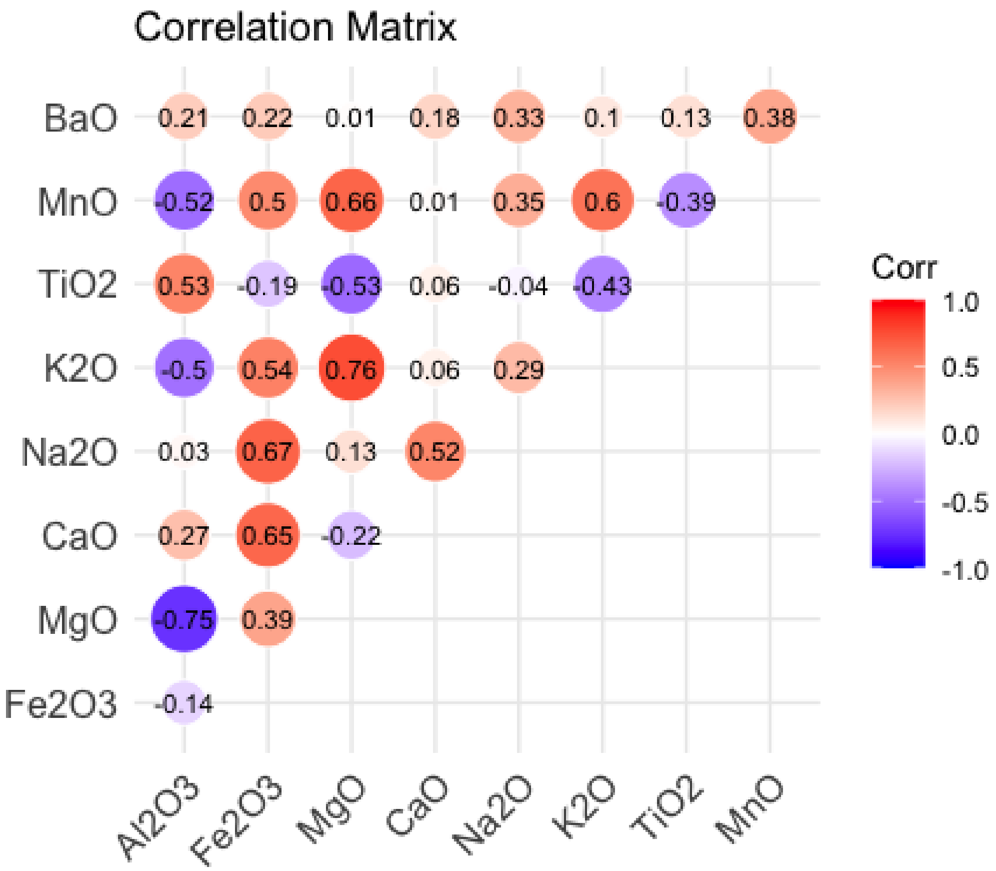
```
# Step2: Piers correlation between independent variables
cor_matrix <- cor(pottery[,4:12], use = "complete.obs")
print(cor_matrix) #explore the output

##             Al2O3     Fe2O3         MgO          CaO        Na2O
K2O
## Al2O3  1.00000000 -0.1399863 -0.745443044  0.266603692  0.03317529 -0.5038
2706
## Fe2O3 -0.13998629  1.0000000  0.387894469  0.649807257  0.67458865  0.5444
1827
## MgO   -0.74544304  0.3878945  1.000000000 -0.216329758  0.13114454  0.7582
```

```
3884
## CaO    0.26660369  0.6498073 -0.216329758  1.000000000  0.52406945  0.0599
4636
## Na2O   0.03317529  0.6745887  0.131144541  0.524069447  1.00000000  0.2861
0997
## K2O   -0.50382706  0.5444183  0.758238837  0.059946364  0.28610997  1.0000
0000
## TiO2   0.52799863 -0.1857044 -0.531258752  0.055893196 -0.03736357 -0.4295
0788
## MnO   -0.52087066  0.5020333  0.660174956  0.005195259  0.34534824  0.6003
3099
## BaO    0.21030295  0.2208851  0.008566432  0.183832222  0.33142159  0.0988
8138
##              TiO2          MnO         BaO
## Al2O3  0.52799863 -0.520870656 0.210302954
## Fe2O3 -0.18570436  0.502033303 0.220885051
## MgO   -0.53125875  0.660174956 0.008566432
## CaO    0.05589320  0.005195259 0.183832222
## Na2O  -0.03736357  0.345348236 0.331421587
## K2O   -0.42950788  0.600330986 0.098881378
## TiO2   1.00000000 -0.393156893 0.128468351
## MnO   -0.39315689  1.000000000 0.381689130
## BaO    0.12846835  0.381689130 1.000000000
```

```
#Plot the correlation as a heatmap
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
ggcorrplot(cor_matrix, method = "circle", type = "upper", lab = TRUE, lab_siz
e = 3, outline.col = "white", colors = c("blue", "white", "red"),title = "Cor
relation Matrix", ggtheme = theme_minimal())
```



What does the resulting correlation matrix tell us?

Let us perform now the PCA

```
# Step 3: Perform PCA
pca_result <- prcomp(pottery[,4:12], center=TRUE, scale. = FALSE)
```

IMPORTANT!: as you can see in the script above, we typed "FALSE" in the argument `scale.`. This is because in our dataset the variables are percentages and so have same units. Instead, in scenarios where either variables have different units (e.g., weights in grams and Porosity in %) or vastly different ranges (e.g., Diameter (10-50 cm) vs decoration complexity (1-5)) scaling is recommended (`scale. = TRUE`). Scaling standardizes numeric variables so that they contribute equally to the PCA analysis.

Let us have a look at the results of our PCA analysis:

```
print(pca_result$rotation)  # View the loadings of each variable

##                   PC1          PC2          PC3          PC4           PC
5
## Al2O3 -7.711701e-01  0.4329847678  0.4621086452  0.0582434318 -1.950209e-0
2
## Fe2O3  3.565279e-01  0.8812062304 -0.2382724970  0.0996716986  1.605671e-0
1
```

```
## MgO    4.819963e-01 -0.0052243824  0.7533657166  0.4114981429 -1.646667e-0
1
## CaO   -8.050823e-03  0.1461829838 -0.1713888029 -0.0822315163 -9.501695e-0
1
## Na2O   1.121140e-02  0.0490640605 -0.0202092417 -0.0006275583 -7.330564e-0
3
## K2O    2.099490e-01  0.1099744028  0.3632584757 -0.9001556162  3.201508e-0
2
## TiO2  -3.735264e-02  0.0080198159 -0.0174621342  0.0172805120  2.046167e-0
1
## MnO    1.413753e-02  0.0060715639  0.0088004006  0.0011692384  3.088534e-0
2
## BaO   -6.893536e-05  0.0004228987  0.0005345961  0.0000301048  1.382099e-0
6
##                PC6          PC7          PC8           PC9
## Al2O3 -0.021899058  0.002217167  0.004369102 -5.794025e-04
## Fe2O3 -0.034856300  0.050460466 -0.010582108  4.769378e-04
## MgO    0.059540776 -0.006289393 -0.007860548 -2.594574e-04
## CaO    0.194303868  0.022571063  0.036633435 -1.409242e-03
## Na2O   0.038712488 -0.995163852 -0.071719830 -3.315929e-03
## K2O    0.022887809  0.002071520 -0.006089359 -2.530077e-05
## TiO2   0.976929822  0.035339048  0.020572354 -2.231306e-03
## MnO   -0.024265023 -0.072550712  0.995865841 -3.360621e-02
## BaO    0.001787553 -0.005654995  0.033351520  9.994259e-01
```

The results show us the *loadings* that represent how much each original variable contributes to the principal components. If we square the values of the *loadings*, we can assess to what degree the new components account for the variation in each variable.

```
pca_result$rotation^2
```

```
##                PC1          PC2          PC3          PC4          PC5
## Al2O3 5.947033e-01 1.874758e-01 2.135444e-01 3.392297e-03 3.803315e-04
## Fe2O3 1.271121e-01 7.765244e-01 5.677378e-02 9.934447e-03 2.578180e-02
## MgO   2.323204e-01 2.729417e-05 5.675599e-01 1.693307e-01 2.711513e-02
## CaO   6.481575e-05 2.136946e-02 2.937412e-02 6.762022e-03 9.028221e-01
## Na2O  1.256954e-04 2.407282e-03 4.084134e-04 3.938294e-07 5.373717e-05
## K2O   4.407859e-02 1.209437e-02 1.319567e-01 8.102801e-01 1.024966e-03
## TiO2  1.395220e-03 6.431745e-05 3.049261e-04 2.986161e-04 4.186799e-02
## MnO   1.998698e-04 3.686389e-05 7.744705e-05 1.367119e-06 9.539045e-04
## BaO   4.752084e-09 1.788433e-07 2.857929e-07 9.062992e-10 1.910197e-12
##                PC6          PC7          PC8          PC9
## Al2O3 4.795687e-04 4.915829e-06 1.908905e-05 3.357073e-07
## Fe2O3 1.214962e-03 2.546259e-03 1.119810e-04 2.274697e-07
## MgO   3.545104e-03 3.955647e-05 6.178821e-05 6.731812e-08
## CaO   3.775399e-02 5.094529e-04 1.342009e-03 1.985963e-06
## Na2O  1.498657e-03 9.903511e-01 5.143734e-03 1.099539e-05
## K2O   5.238518e-04 4.291194e-06 3.708030e-05 6.401290e-10
## TiO2  9.543919e-01 1.248848e-03 4.232218e-04 4.978727e-06
## MnO   5.887914e-04 5.263606e-03 9.917488e-01 1.129377e-03
## BaO   3.195344e-06 3.197897e-05 1.112324e-03 9.988520e-01
```

For instance, the new component PC1 accounts for *59 %* of the variation in the variable *Aluminium oxide (Al2O3)*

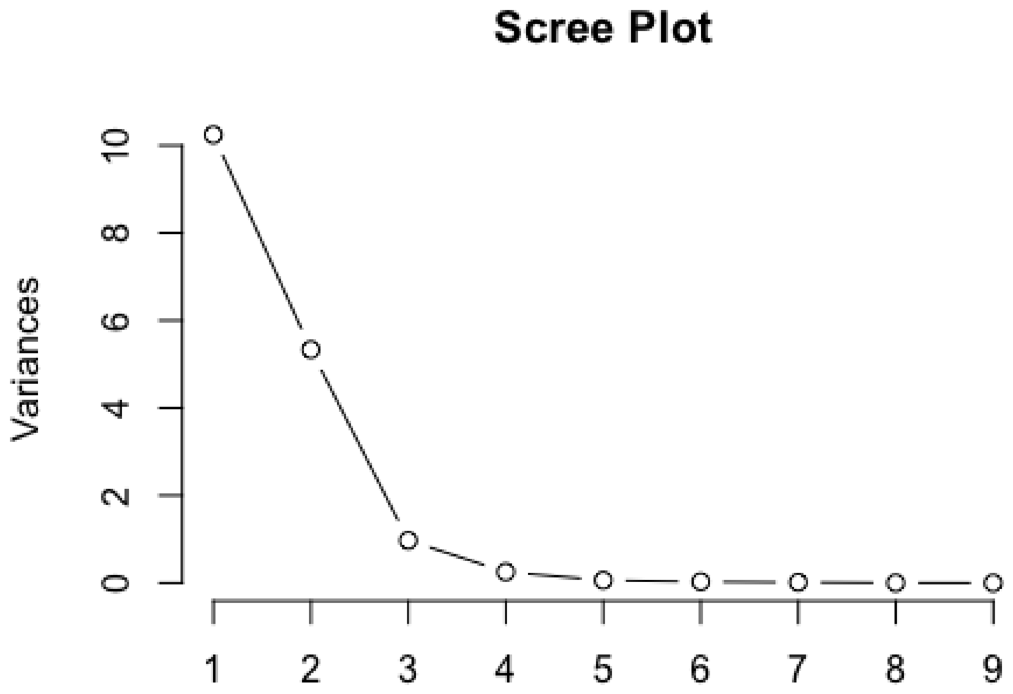
Let us now check the results of the PCA:

```
summary(pca_result)

## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC
7
## Standard deviation     3.2012 2.3099 0.9863 0.50310 0.24965 0.17165 0.1258
6
## Proportion of Variance 0.6057 0.3154 0.0575 0.01496 0.00368 0.00174 0.0009
4
## Cumulative Proportion  0.6057 0.9211 0.9786 0.99352 0.99721 0.99895 0.9998
8
##                            PC8      PC9
## Standard deviation     0.04439 0.002353
## Proportion of Variance 0.00012 0.000000
## Cumulative Proportion  1.00000 1.000000
```

The results above tell us that PC1 explains **60.57 %** of the variation in the original *nine* variables, while PC2 explains **31.54 %** of the variation. The Cumulative proportion informs us that PC1 and PC2 together explains **92.11 %** of the variation.

After performing PCA, a key decision is determining the number of principal components to retain. A common guideline is to select enough components so that they collectively explain at least 90% of the variance in the data. Alternatively, one can identify the optimal number of components by examining the point where there is a noticeable change in the slope of the scree plot.

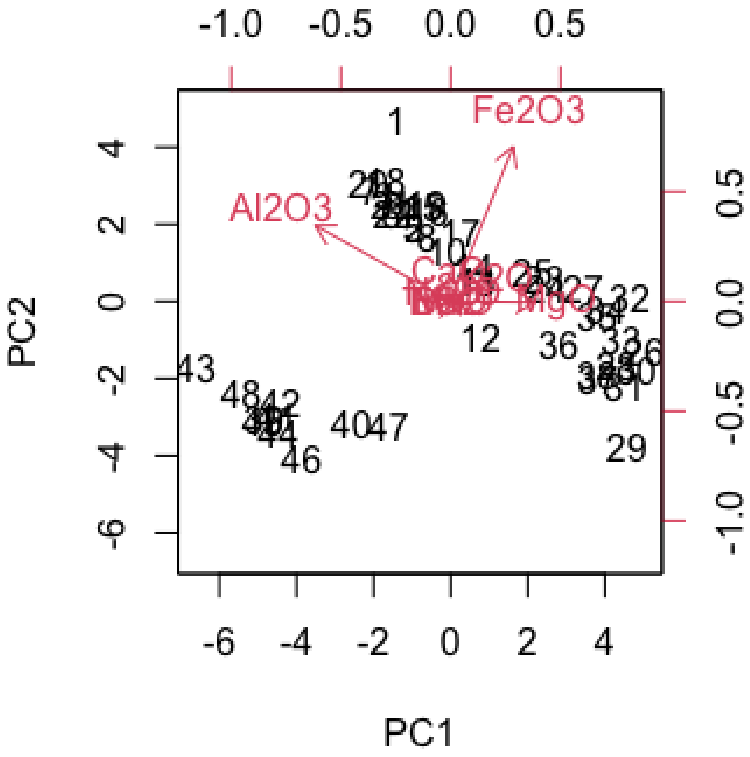
```
# Step 4: Visualize PCA Results
# Scree Plot (Explained Variance)
plot(pca_result, type = "l", main = "Scree Plot")
```



The “elbow” point on the scree plot indicates the number of principal components that explain most of the variance in the data. Components before the elbow are usually retained, while those after the elbow contribute little new information and can be discarded. So, in this case we retain 2 principal component.

Let us visualise a biplot:

```
# Step5:Biplot to visualize PCA components
biplot(pca_result, scale = 0)

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], lengt
h =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
```



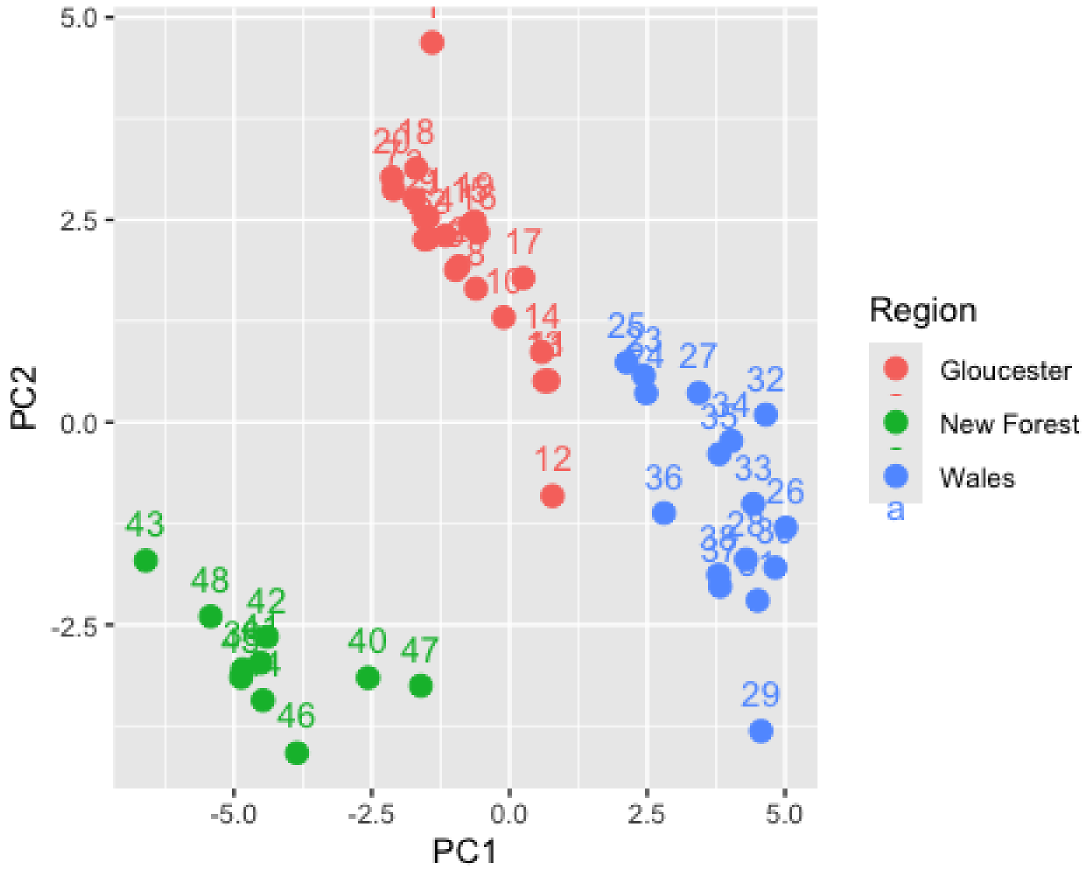
A biplot in the context of Principal Component Analysis (PCA) is a graphical representation that helps visualize both the scores (the transformed data in the principal component space) and the loadings (the weights or contributions of the original features to the principal components) on the same plot.

These represent how much each original feature (variable) contributes to each principal component. The loadings are shown as vectors (arrows) on the plot. The length of the arrow indicates the magnitude of the contribution of that feature to the principal component. The direction of the arrow indicates the feature's relationship with the principal component. If the arrow points in a similar direction to the axis, that feature has a strong positive correlation with that principal component; if it points in the opposite direction, the feature has a strong negative correlation.

Finally, let us plot a PCA scatterplot:

```
# Step 6: Create a PCA DataFrame for ggplot
pca_scores <- as.data.frame(pca_result$x)
pca_scores$Region <- pottery$Region  # add the column "Region" as a categoric
al variable
pca_scores$Id <- pottery$Id # add the column "Id"

# Scatterplot of PC1 vs PC2
library(ggplot2)
```

```r
ggplot(pca_scores, aes(x = PC1, y = PC2, color = Region)) +
  geom_point(size = 3) +
  geom_text(aes(label = Id), vjust = -1, size = 4)

theme_minimal()
```



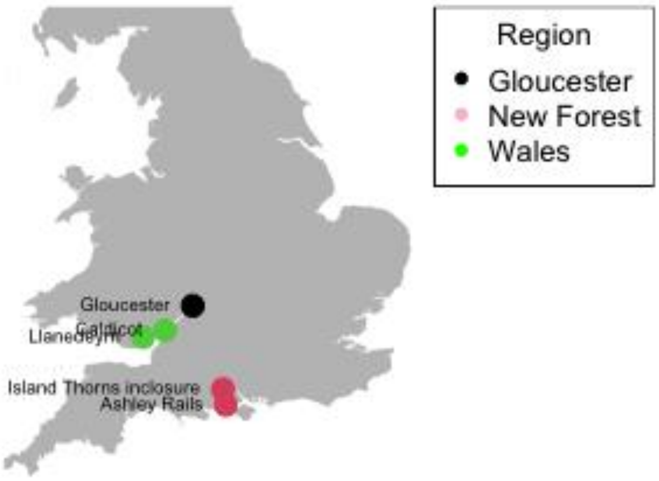
You can see that PCA has grouped our pottery into three distinct groups. In addition, we can see that the groups match with the regions of provenance of pottery. This has deep implications both in terms of production techniques and material sources.

Given the results above, let us plot the distibution of the archaeological sites from which the pottedy data come from. Load the vector data:

```r
library(sf) #Load required package
uk <- st_read("PCA/uk.shp")

sites<-st_read("PCA/sites.shp")
```

Plot the distribution of sites yielding Roman pottery:

```
plot(st_geometry(uk), col="grey75", border=NA,  xlim=c(-4, 1), ylim=c(50, 55))
points(st_coordinates(sites), col = as.factor(sites$region), pch = 19, cex = 1.3) # add sites coloured by region
text(st_coordinates(sites), labels = sites$name, pos = 2, cex = 0.5, col = "black") #add labels
legend("topright", legend = c("Gloucester", "New Forest","Wales"), cex=0.8, col = c("black","pink","green"), pch = 16, title = "Region")
```



You have done also with today's tutorial!

References

- Hodder, I., and Orton, C., 1976. *Spatial analysis in archaeology*. Cambridge: Cambridge University Press.
- Tubb, A., A. J. Parker, and G. Nickless. 1980. The Analysis of Romano-British Pottery by Atomic Absorption Spectrophotometry. *Archaeometry 22*: PP. 153-71